N – Modularization and Utility Classes

[DRAFT - ~95% Complete]

# LOG Review

These are some very brief points regarding the concepts and skills listed in the LOGs for this topic.

# Overview

This topic examines the technique of modularization as a way to simplify complex logic. In addition, this topic looks at the use of “utility” classes to aid in simplifying code and enabling better “code-reuse” for repetitive tasks. This technique of “simplifying” complex logic is also known as “refactoring”.

Modularization will be examined in the context of simplifying the internal working of a class’ methods and in the context of organizing and simplifying a driver. For classes, modularization is simply taking any repetitive or overly-complex code out of the class’ existing methods and placing that code inside of private methods. For console drivers (which typically consist of static methods),

…. The following keywords are introduced. (Note that additional keywords from previous topics may also be present.)

* Keyword

This topic will introduce the following grammars, syntax and language constructs. (Note that additional concepts from previous topics may also be present.)

* Grammer
* Syntax
* Language Aspect

This topic will also (more notes…).

**Daily LOGs**

The following daily LOGs are covered in this package.

**OOP Basics**

* Object-centric LOG….
  + Overloading constructors

**General Programming Concepts and Terms**

* General logs:
  + Distinguish between Variables, Values and Data Types

# Notes

# Examples

# Exercises

# Solutions